



DATASTREAM

2

Request Packet from VT Server to host system

0050000000000000054376200033111000001111000000010000Y

Response Packet from host system to VT Server

005800000000000000543762033111000001111000+000075000+0000725

Fig. 1



APPLICATION PROGRAMMING INTERFACE

REQUEST PACKET

FIELD DESCRIPTION	FORMAT	CONTENT
TRANSACTION CODE	3N	'033'
CREDIT UNION ACCESS CODE	3N	CODE ASSOCIATED WITH EACH CREDIT UNION ASSIGNED BY HOST
MEMBER NUMBER TO WITHDRAW FUNDS FROM	9N	(ENTERED BY CALLER)
ACCOUNT SUFFIX TO WITHDRAW FUNDS FROM	3N	(ENTERED BY CALLER)
TRANSFER AMOUNT	9N	(TWO DECIMAL POSITIONED ASSUMED)
POST INDICATOR	1A	N - PRELIMINARY EDIT DO NOT UPDATE FILES
		Y - MEMBER HAS CONFIRMED THEY WANT TO POST TRANSACTION UPDATE FILES

RESPONSE PACKET

FIELD DESCRIPTION	FORMAT	CONTENT
TRANSACTION CODE	3N	'033'
CREDIT UNION ACCESS CODE	3N	
MEMBER NUMBER	9N	
HOST RESPONSE CODE	3N	000 - POSITIVE RESPONSE, CONTINUE SCRIPT
		210-214 - READ ERROR, REPEAT MENU
		220-221 - READ ERROR, REPEAT MENU
SIGN FIELD	1A	+ OR -, NEGATIVE OR POSITIVE BALANCE
CURRENT BALANCE OF WITHDRAWAL FROM ACCOUNT (BEFORE TRANSFER)	9N	N - PRELIMINARY EDIT DO NOT UPDATE FILES
SIGN FIELD	1A	+ OR -, NEGATIVE OR POSITIVE BALANCE
AVAILABLE BALANCE OF WITHDRAWAL FROM ACCOUNT (BEFORE TRANSFER)	9N	

FIG. 2



REQUEST PACKET

MAPPING DOCUMENT

API FIELD DESCRIPTION	VT SERVER FIELD DEFINITIONS
TRANSACTION CODE	HARD CODE - SET TO '033'
CREDIT UNION ACCESS CODE	CODE RETRIEVED FROM DATABASE CONFIGURATION FOR EACH PARTICULAR CREDIT UNION
MEMBER NUMBER TO WITHDRAW FUNDS FROM	FROM INTERNAL FIELD TRANSACACCNFROM BEFORE FIELD SEPARATOR
ACCOUNT SUFFIX TO WITHDRAW FUNDS FROM	FROM INTERNAL FIELD TRANSACACCNFROM AFTER FIELD SEPARATOR
TRANSFER AMOUNT	FROM INTERNAL FIELD TRANSACAMOUNT. INTERNAL FIELD INCLUDES DECIMAL POINT. EXTERNAL FIELD DOES NOT. REMOVE DECIMAL POINT BEFORE SENDING. MAXIMUM AMOUNT IS 9,999,999.99
POST INDICATOR	SET BASED ON TRANSACPOSTMODE. IF 0, SET TO N. IF 1, SET TO Y.

RESPONSE PACKET

FIELD DESCRIPTION	CONSTANT
TRANSACTION CODE	FIELD IS ECHOED, NOT USED IN RESPONSE
CREDIT UNION ACCESS CODE	FIELD IS ECHOED, NOT USED IN RESPONSE
MEMBER NUMBER	FIELD IS ECHOED, NOT USED IN RESPONSE
HOST RESPONSE CODE	EXTERNAL SYSTEM'S RESPONSE CODE. MAP TO THE VT SERVER RESPONSE CODE BASED ON THE CONFIGURATION TABLES.
SIGN FIELD	POSITIVE/NEGATIVE SIGN INDICATOR FOR FIELD THAT FOLLOWS. USE TO MAP APPROPRIATELY.
CURRENT BALANCE OF WITHDRAWAL FROM ACCOUNT (BEFORE TRANSFER)	MAP TO THE LEDGER BALANCE FIELD OF THE INTERNAL MESSAGE. QBT BALANCE INDICATOR FOR LEDGER IS 1.
SIGN FIELD	POSITIVE/NEGATIVE SIGN INDICATOR FOR FIELD THAT FOLLOWS. USE TO MAP APPROPRIATELY.
AVAILABLE BALANCE OF WITHDRAWAL FROM ACCOUNT (BEFORE TRANSFER)	MAP TO THE LEDGER BALANCE FIELD OF THE INTERNAL MESSAGE. QBT BALANCE INDICATOR FOR LEDGER IS 2.

PRIOR ART

FIG. 3

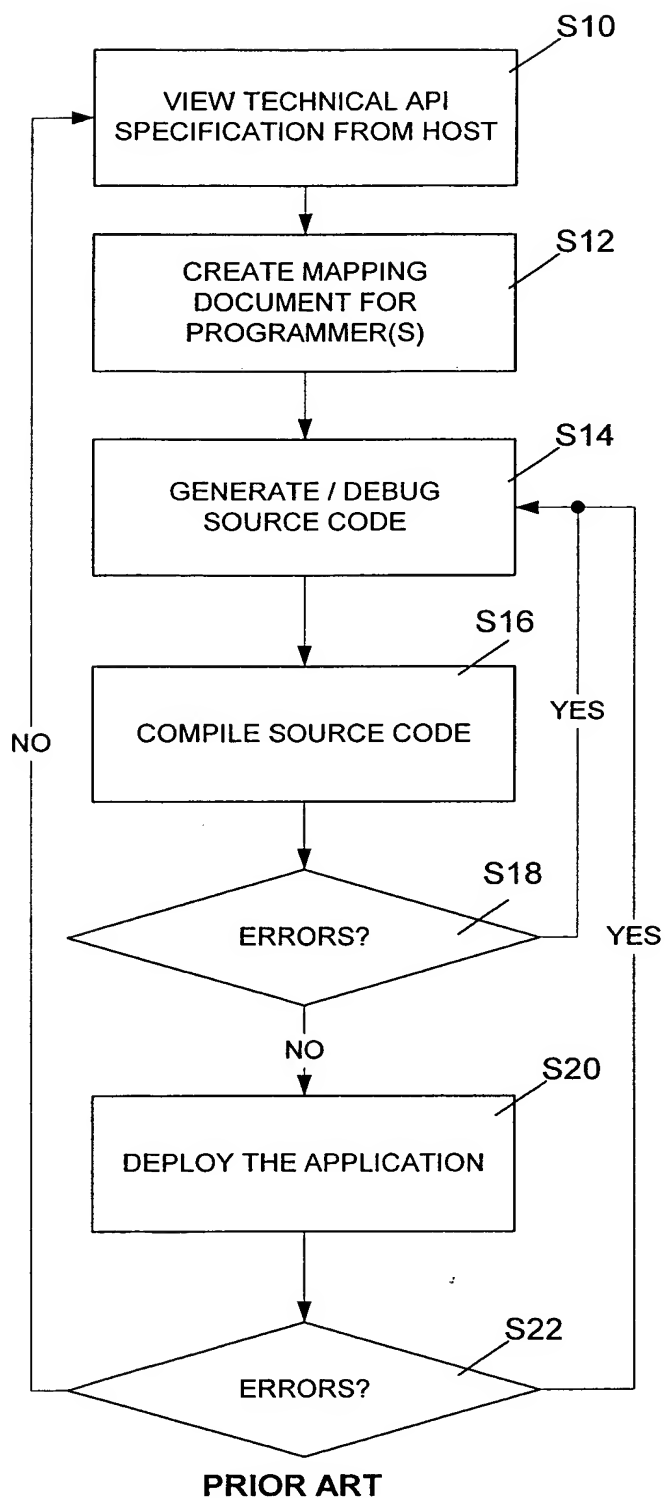


Fig. 4

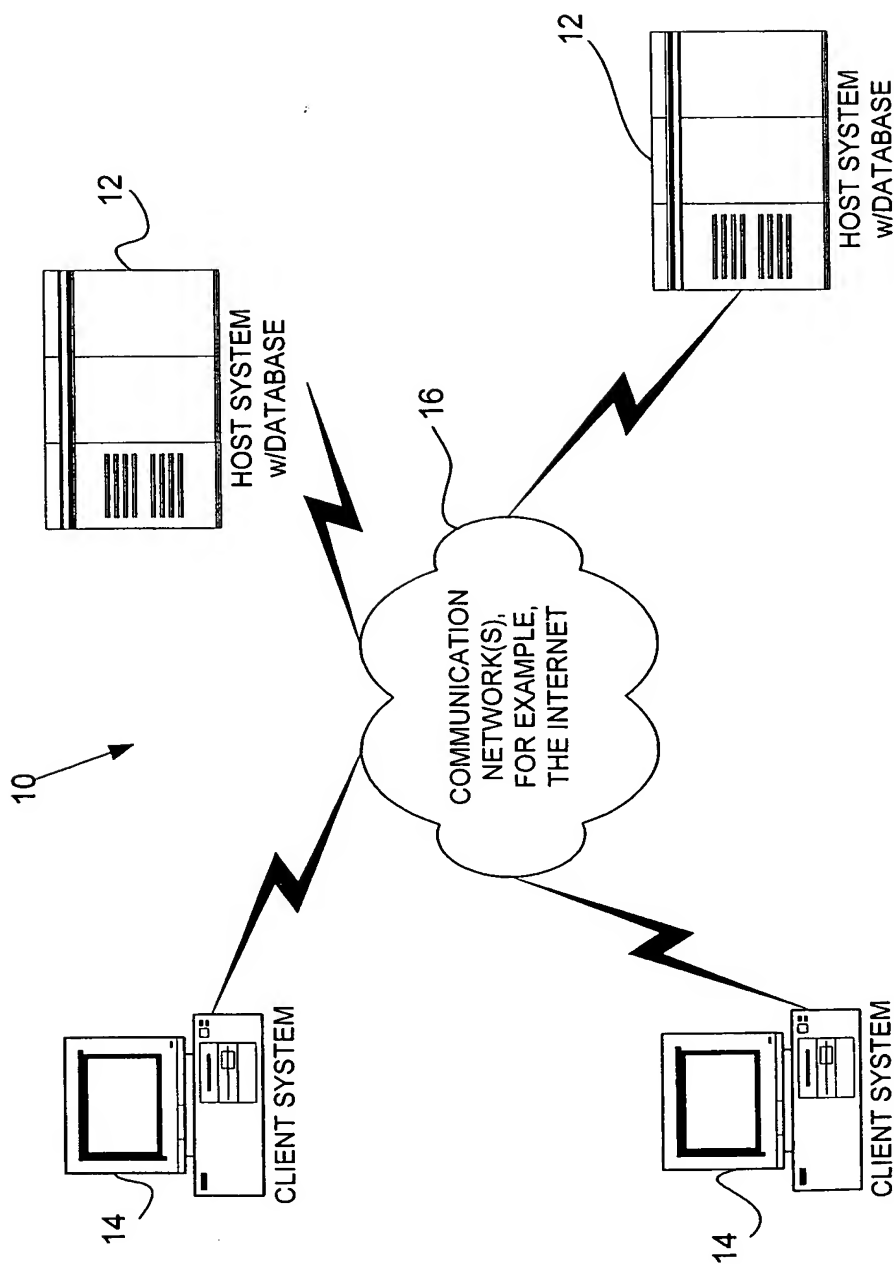
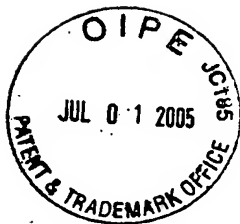


Fig. 5

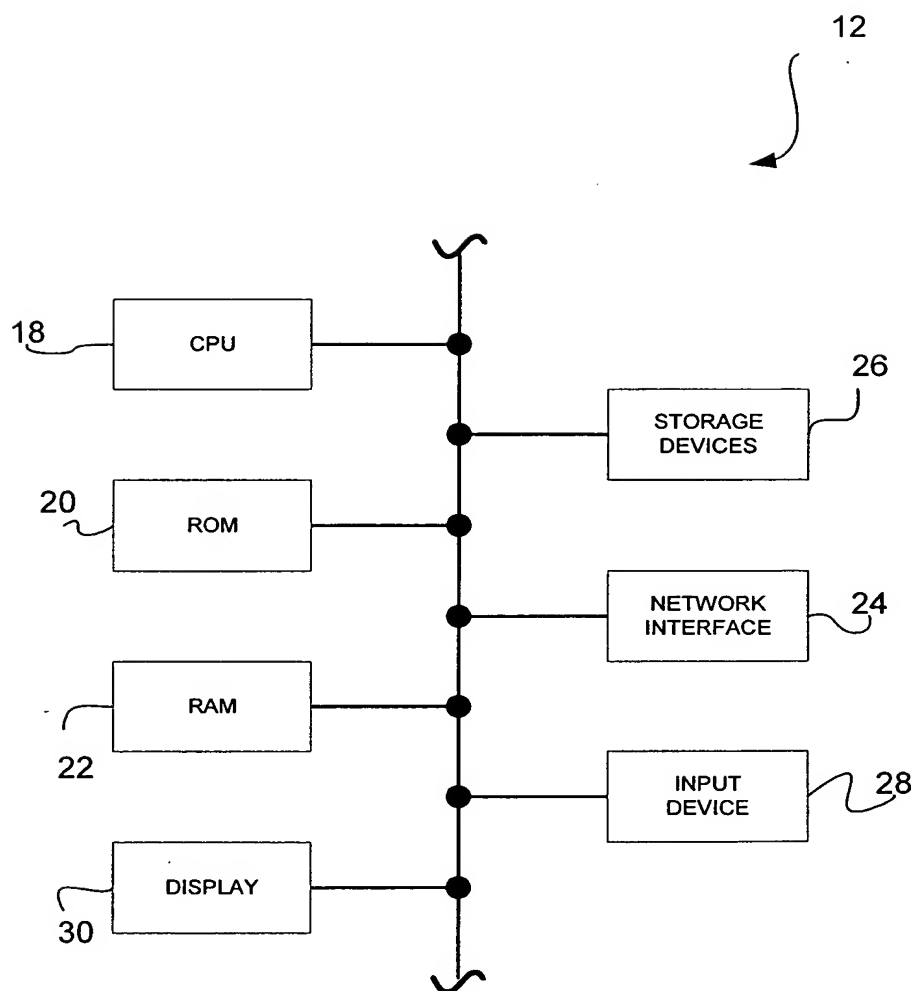


Fig. 6

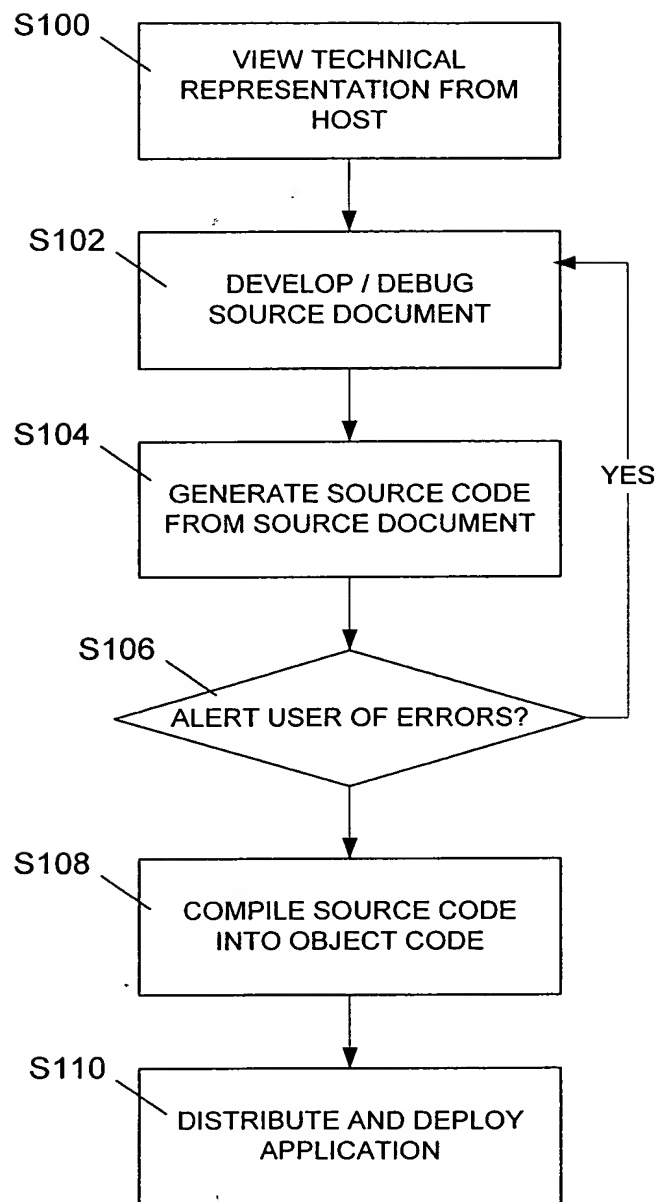


Fig. 7

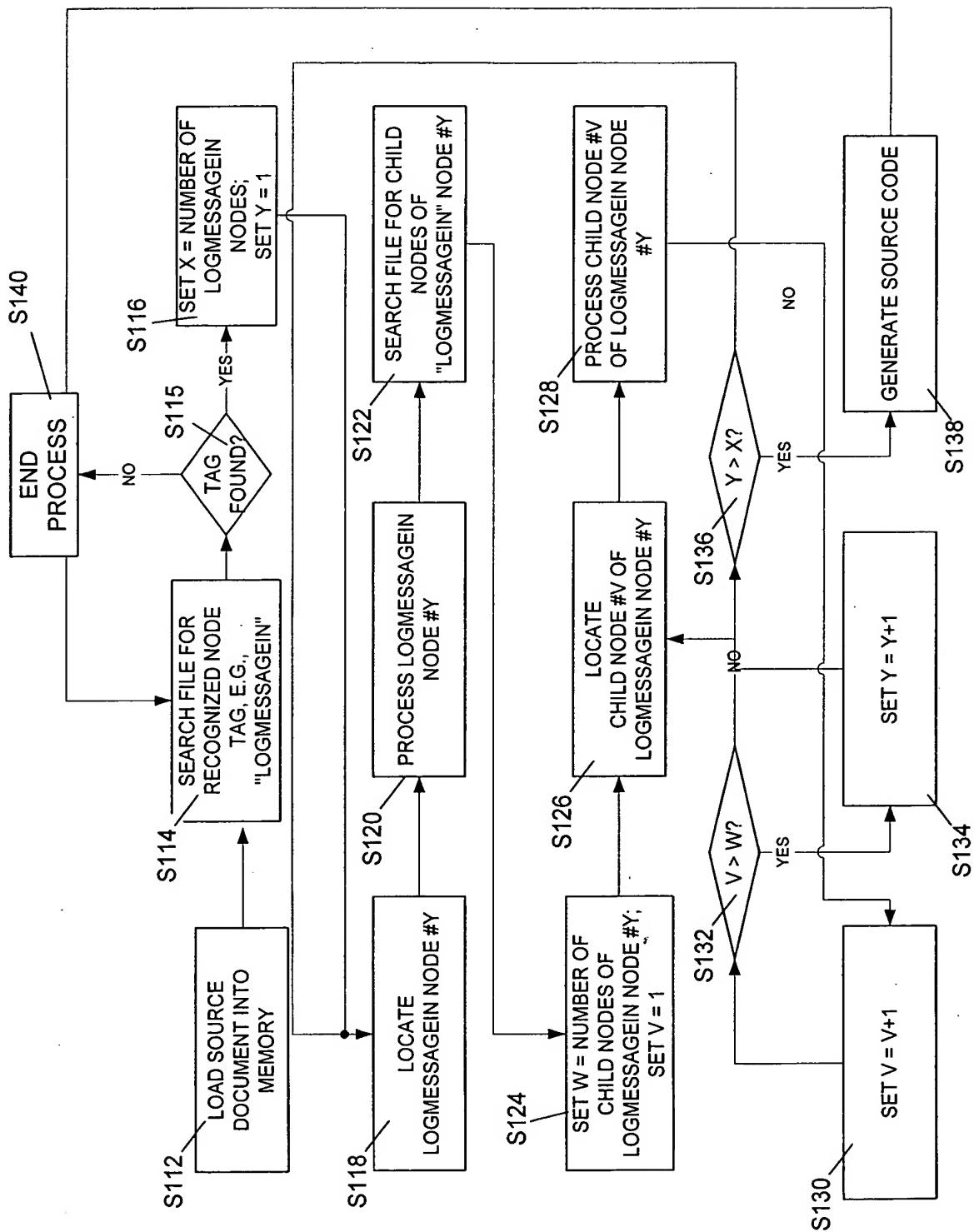


Fig. 8



```
1. if ( !FAILED( spElement->getElementsByTagName(  
2.     L"LogMessageIn", &spNodes ) ) && spNodes )  
3. {  
4.     long lLength = 0;  
5.     if ( FAILED( spNodes->get_length(&lLength)) )  
6.     {  
7.         return ;  
8.     }  
9.     CString sText;  
10.    for ( long i = 0 ; i < lLength ;i++ )  
11.    {  
12.        spNode.Release();  
13.        if ( !FAILED ( spNodes->get_item( i , &spNode ) ) )  
14.        {  
15.            CheckOutputLine( 0, spNode );  
16.            GetAttribute( spNode , L"id", sText );  
17.            if ( m_bDoVB )  
18.            {  
19.                sText = _T("Sub LogMessageIn_")+ sText + _T("( vtObj as VTMsgObj )");  
20.                OutputLine( 0 , sText );  
21.                OutputLine( 0, _T("\r\n") );  
22.                OutputLine( 1, _T("Temp variables used by routine\r\n"));  
23.                OutputLine( 1, _T("Dim sTmp As String\r\n") );  
24.                OutputLine( 1, _T("Dim sTmp4 As String\r\n") );  
25.                OutputLine( 1, _T("Dim sTmp5 As String\r\n") );  
26.                OutputLine( 1, _T("Dim sTmp3 As String\r\n") );  
27.                OutputLine( 1, _T("Dim sTmp2 As String\r\n") );  
28.                OutputLine( 1, _T("Dim sCmp As String\r\n") );  
29.                OutputLine( 1, _T("Dim iOffset As integer\r\n") );  
30.                OutputLine( 1, _T("\r\n") );  
31.                OutputLine( 1, _T("On Error Goto ErrOut\r\n") );  
32.                OutputLine( 1, _T("\r\n") );  
33.            } else  
34.            {  
35.                AddCFunction( "LogMessageIn_" + sText);  
36.                sText = _T("void MsgHandler::LogMessageIn_" ) + sText;  
37.                sText += _T("( IDualVTMsgObj * vtObj )\r\n{\r\n\ttry {\r\n");  
38.                OutputLine( 0 , sText );  
39.                OutputLine( 0, _T("\r\n") );  
40.                OutputLine( 1, _T("//Temp variables used by routine\r\n"));  
41.                OutputLine( 1, _T("CComBSTR sTmp;\r\n") );  
42.                OutputLine( 1, _T("CComBSTR sTmp2;\r\n") );  
43.                OutputLine( 1, _T("CComBSTR sTmp3;\r\n") );  
44.                OutputLine( 1, _T("CComBSTR sTmp5;\r\n") );  
45.                OutputLine( 1, _T("CComBSTR sTmp4;\r\n") );  
46.                OutputLine( 1, _T("CComBSTR sCmp;\r\n") );  
47.                OutputLine( 1, _T("int iOffset = 0, iLastPos = 0;\r\n") );  
48.                OutputLine( 1, _T("\r\n") );  
49.                OutputLine( 1, _T("//End of temp variables\r\n"));  
50.                OutputLine( 1, _T("\r\n") );  
51.            }  
  
52.            m_bProcessingGetResponseCode += 1;  
53.            ProcessMessageIn( 1, spNode );  
54.            m_bProcessingGetResponseCode -= 1;  
  
55.            if ( m_bDoVB )  
56.            {  
57.                OutputLine(0, _T("ErrOut:\r\n"));  
58.                OutputLine(1, _T("vtObj.EndSetError Err.Number, Err.Description\r\n"));  
59.            }
```

Fig. 9

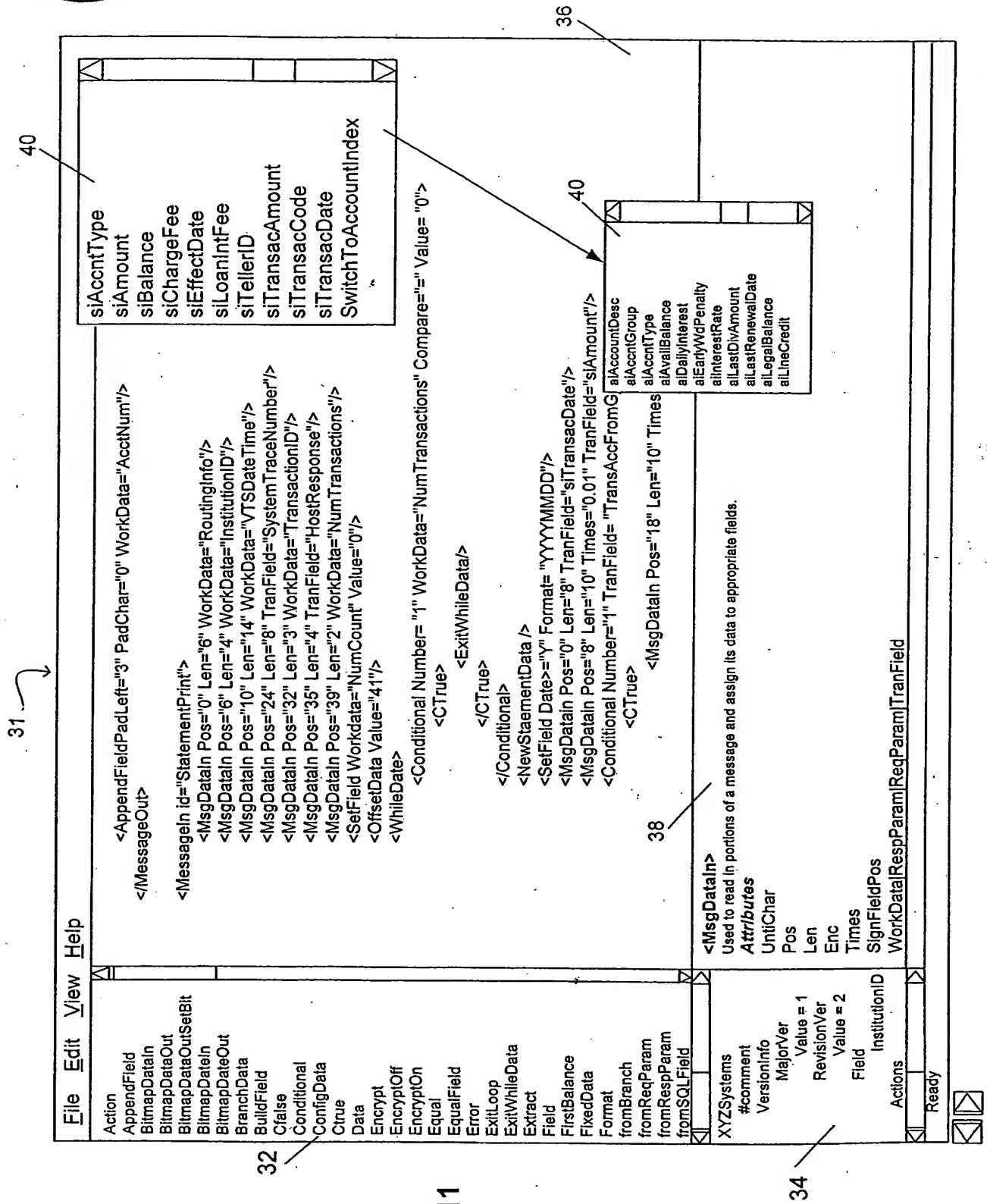


FIG. 11